

专 利 合 作 条 约

PCT

专利性国际初步报告

(PCT 第II章)

(PCT 36 和细则 70)

REC'D 05 AUG 2005

WIPO


PCT

申请人或代理人的档案号 FPCL04150011	关于后续行为 参见 PCT/IPEA/416 表	
国际申请号 PCT/CN2004/000375	国际申请日(日/月/年) 19. 04 月 2004 (19.04.2004)	优先权日(日/月/年) 17.04 月 2003 (17.04.2003)
国际专利分类(IPC)或者国家分类和 IPC 两种分类 IPC7 G06F7/00		
申请人 李志中, 徐菊园		

1. 本报告是国际初步审查单位根据条约 35 做出的国际初步审查报告, 并依照条约 36 将其传送给申请人。
2. 本报告共计 3 页, 包括扉页。
3. ☐ 本报告还有附件,
 - a. ☐ (传送给国际局和申请人)共计 ____ 页, 包含
 - ☐ 修改后的并且作为本报告基础的说明书修改页、权利要求书修改页和/或附图修改页, 和/或对本国际初步审查单位所做出的更正页(见 PCT 细则 70.16 和行政规程 607)。
 - ☐ 国际初步审查单位认为修改超出原始公开范围的取代页, 参见第 I 栏第 4 项和补充栏。
 - b. ☐ (传送给国际局) 共计 (指明电子载体的类型和数量) ____, 包含有在与序列表有关的补充栏中指明的电子形式的序列表和/或与其相关的表格。(行政规程 802)

4. 本报告包括关于下列各项的内容:

- I ☒ 报告的基础
- II ☐ 优先权
- III ☐ 不做出关于新颖性、创造性和工业实用性的意见
- IV ☐ 缺乏发明的单一性
- V ☒ 按条约 35(2)关于新颖性、创造性或工业实用性的理由; 支持这种意见的引证和解释
- VI ☐ 引用的某些文件
- VII ☐ 国际申请中的某些缺陷
- VIII ☐ 对国际申请的某些意见

提交要求书的日期 01.09 月 2004 (01.09.2004)	完成本报告的日期 04.07 月 2005 (04.07.2005)
中华人民共和国国家知识产权局 IPEA/CN 中国北京市海淀区西土城路 6 号(100088) 传真号: (86-10)62019451	受权官员 A413 电话号码 (86-10): 62084990 

I. 报告的基础

1. 关于语言, 本报告将基于:

☒ 申请提出时使用的语言。

☐ 该申请的____语言译文, 提供该种语言的译文是

☐ 为了国际检索而提交的译文所使用的语言(细则 12.3 和 23.1 (b))。

☐ 为了国际申请的公布而提交的译文所使用的语言(细则 12.4)。

☐ 为了国际初步审查而提交的译文所使用的语言(细则55.2和/或55.3)。

2. 关于国际申请中各个部分, 本报告基于(申请人为答复受理局根据条约 14 所发通知而提交的替换页, 在本报告中视为“原始提交”的文件, 不作为本报告的附件)

☐ 原始提交的国际申请。

☒ 说明书, 第____页 原始提交的,
第 1-25 页 2004 年 9 月 1 日 初审单位收到的,
第____页 初审单位收到的。

☒ 权利要求, 第____页, 原始提交的,
第____页, 按条约 19 条修改的(附有说明),
第 26-29 页 2004 年 9 月 1 日 初审单位收到的,
第____页 初审单位收到的。

☒ 附图, 第 1 页, 原始提交的。
第 2 页*, 2004 年 9 月 1 日 初审单位收到的,
第____页*, 初审单位收到的。

☐ 序列表和/或相关表格——参见与序列表有关的补充栏。

3. 修改导致以下内容的删除:

☐ 说明书, 第____页
☐ 权利要求, 第____项
☐ 附图, 第____页, 图____
☐ 序列表(具体说明)____
☐ 与序列表相关的表格(具体说明)____

4. ☐ 由于本报告附件的(某些)修改, 如下所列, 被认为超出了原始公开的范围, 如补充栏所示, 因此本报告是按照没有修改的情况做出的(细则 70.2(c))。

☐ 说明书, 第____页
☐ 权利要求, 第____项
☐ 附图, 第____页, 图____
☐ 序列表(具体说明)____
☐ 与序列表相关的表格(具体说明)____

*如果第 4 项适用, 一些或全部的文件页可能做出“被取代”标记。

V. 按条约 35 (2) 关于新颖性、创造性或工业实用性的意见；支持这种理由的引证和解释

1. 意见

新颖性(N)	权利要求 1—13	是
	权利要求	否
创造性(IS)	权利要求 1—13	是
	权利要求	否
工业实用性(IA)	权利要求 1—13	是
	权利要求	否

2. 引证和解释 (细则 70.7)

(1) 参考以下文件:

D1:US6073149A

D2:CN1136680A

(2) 本发明涉及数字工程方法和处理器领域，提出一种新的数字工程方法，提高运算速度。本发明的混 Q 进制、进位行数字工程方法包括：

将参与运算的普通 Q 进制数的每一位数字都加上一个数符，参予运算的数为 k 个混 Q 进制数，对 k 个数同时进行混 Q 进制的求和运算，从最低位开始，按位相加，即在某一位上，取前述 k 个数中的二个数按位相加，得到“按位和”，将此和数记入下一运算层，作为“部份和”数，同时所得“混数进位”，则存放到下一运算层中的任一进位行中与该位相邻的高位处，经过对此运算，直至不产生“混 Q 进位”为止，则最后一次“按位加”所得和数，即为所求加法运算结果。本发明提供了混 Q 进制、进位行处理器。

(3) D1 公开了一种多值加法器计算电路，包括并行加法单元，输出加法单元，量化单元和逻辑转换单元。所述的并行加法单元计算所述两个多值数字相应位和进位的中间和。所述加法单元将各个并行加法单元输出的一个低位位的进位输出同该位的“中间和”输出相加而生成该位的多值输出的设置了运算位数。

D2 公开了一种利用改进的布斯算法有选择地执行无符号数值乘法或有符号数值乘法来进行乘法操作的乘法器。它包括给各个输入端提供扩展位以便在用二进制补码格式表示的有符号数值乘法中执行无符号数值乘法的选择单元，还包括执行被符号位增强的符号数字操作的部分积生成器。它最好还包括产生和传送先行进位的先行进位加法器。

(4) 权利要求 1—13 满足 PCT 第 32 款第 (2) — (4) 项规定的标准。因为，本方法和处理器处理的是混 Q 进制数。在操作之前，参加运算的普通 Q 进制数的每一位数字都加上一个数符。现有技术进行了按位加，但没有使用混 Q 进制和“对冲”和“划十”机制。

01 · 9月 2004 (01 · 09 · 2004)

China Patent & Trademark
Agent (U.S.A.) LTD.
55 Broad Street, 15th Floor,
New York, NY 10004, U.S.A.
Tel: (1-212)8098100

Tokyo Office
Room 1003, Bureau Toranomon
2-7-16 Toranomon Minato-ku,
Tokyo 105-0001, Japan
Tel: (81-3)52511966

CHINA PATENT AGENT (H.K.) LTD.

中国专利代理 (香港) 有限公司

Patent, Trademark, Copyright & Legal Affairs
22/F, GREAT EAGLE CENTRE,
23 HARBOUR ROAD, WANCHAI, HONG KONG.
TELEPHONE: (852)28284688 FACSIMILE: (852)28271018

Munich Office
Zweibrueckenstrasse 17,
D-80331 Munich, Germany
Tel: (49-89)2289328

Beijing Office
B-19/F, Investment Plaza,
27 Jinrong Street,
Xicheng District
Beijing, China 100032
Tel: (86-10)66211588

中国国家知识产权局专利局 PCT 处
北京市海淀区蓟门桥西土城路六号
邮政编码: 100088

(EPLI01)

日期: 2004 年 8 月 31 日

发明名称: 混 Q 进制、进位行数字工程方法和处理器

国际申请号: PCT/CN2004/000375

我方卷号: FPCL04150011P

致中国专利局 PCT 处:

根据 PCT 条约第 34 条, 申请人在请求国际初审时提交了对说明书、权利要求书、说明书摘要以及附图的修改。现提交对说明书、权利要求书和说明书摘要全文的替换页以及说明书附图第 2 页的替换页。谢谢!



来信请寄往: 中国专利代理 (香港) 有限公司北京办事处
北京市西城区金融街 27 号投资广场 B 座 19 层
电话: 010-66211588 邮编: 100032
传真: 010-66214497

混 Q 进制、进位行数字工程方法和处理器

技术领域

本发明涉及数字工程方法和处理器领域，特别是处理器的运算器

背景技术

数字工程包括数控机床、大中型数字化设备和数字系统工程等等。本发明中“数字工程”是专指“数字计算系统工程”。它不是解决一个个具体的算题、或定理证明、或几何问题、或某种数学思想，而是解决四则运算法则等计算系统本身的数字工程实现技术方案。它与具体的计算工具密切相关。众所周知，“计算”有好多种，除“近似计算”、“模拟计算”及“无工具计算”（心算、指算、口算，包括口诀、速算、估算）外，则为“采用工具的数字计算”。“采用工具的数字计算”历史上包括笔算、珠算、机械算、电算，以及筹算等。现代只剩下三种，这就是数字电算、珠算、笔算。与此相应的数字计算工程也就仅有三种：数字计算机；算盘；采用笔和纸进行笔算的数字计算工程，简称为“笔算工程”。

四则运算是数的最基本运算。正如恩格斯所说：“四则（一切数学的要素）。”加法又是四则运算的最基本的运算。因此，我们理所当然应当对四则运算，尤其是对加法运算给予特别的关注。当前数字工程方法中数学的四则运算，首先是加法，有许多不尽如人意之处。主要表现为运算速度慢；在减法中，未能充分利用负数的作用，而且，不能“连减”。尤其在加减混合运算中，不能一步到位；在乘法中，加法的缺点更加扩大严重；在除法中，上述缺点依旧。总之，在最小的数体——有理数体中，四则运算情况并不满意。

在笔算数字工程中，对运算的解剖，表明存在一些隐含的操作程

$$123456+345678=469134$$

$$78+297+259=634$$

$$\begin{array}{r} 123456 \\ + 345678 \\ \hline 469134 \end{array}$$

$$\begin{array}{r} 78 \\ 297 \\ + 259 \\ \hline 634 \end{array}$$

式一

式二

序，以至产生“隐患”。以加法为例，例一“两数相加”，算式如式一。其中，十位上的和数 3，解剖一下，其微程序操作是：（凡未注明所属数制的数，均为普通十进制数。下同。）

①个位上来的进位（见标志）②十位上 5、7 两数字与低位进位相加，即 $(5+7+1)$ 。取其和的个位。③上列 $(5+7+1)$ 和的进位送到高位（见标志）。其余各位情况类似。又如，例二，设三数求和，算式如式二： $78+297+259=634$ 如图可见，上述情况更为加重。显然，存在下列缺点：

a. 进位标示困难。若用小数字表明，则易混淆且字面积受限。特别是表 456789 时就更烦人；若以“.”字写在数字间，则易与小数点混淆且表示 456789 也不便；若以手指数数，则速度慢且不方便；若心算，则费脑力且易错。总之，比较讨厌，易出错。

b. 一般两数相加时，每一位上要有三个数相加求和。于是，需三重运算。三及三以上个数求和时，则更不方便。

c. 验算困难。一般采用重做一遍，费时费力。

减法比加法麻烦。且不能在同一竖式中“连减”，必须断开。特别在加减混合运算时，不能一步到位。

乘除法中，这类情况更为严重。而且，加减乘除运算格式不统一，除法时另起炉灶。

另一方面，在电子计算机的数字工程中，同样有大量的数值运算。这些数一般均采用普通二进制数制{二}来表示。其负数常以原码、反码、补码、移码之类来表示。在现有计算机中运算均以二个数运算，而无法实现“多重运算”。所谓“多重运算”是指多于二个数同时进行加减。

在采用其他普通 Q 进制{Q}等普通数制的电子计算机中，存在相应的许多复杂性。[Q 为自然数。]

发明内容

本发明提出一种新的数字工程方法，显著提高运算速度；同时加强运算正确性的保障，在“笔算工程”中，大大降低笔算的出错率。

本发明的另一个目的是提供一种新的处理器，在现有制造技术的基础上，在设备量相近的情况下，显著提高电子计算机的运算速度。

根据本发明的一个方面，提供一种混 Q 进制、进位行数字工程方法，采用“进位行方法”的“混 Q 进制”运算。包括以下步骤：

第 1 步，设，参予运算的数为 K 个普通 Q 进制数，为 $K \geq 2$ 的正整数， Q 为自然数；将这些普通 Q 进制数的正负符号，分配到相应这些数的每一位上去。

第 2 步，对 K 个数同时进行混 Q 进制的求和运算，从最低位开始或各位同时按位相加，即在某一位上，取前述 K 个数中的二个数按位相加，得到“按位和”为该位这二个数相加的和数，将此和数记入下一运算层，作为“部份和”数；同时所得“混 Q 进位”，则存放到下一运算层的任一进位行中与该位相邻的高位处；

第 3 步，在该位上取 K 个数中的另二个数，进行第 2 步的运算，如此反复，直至 K 个数均取完为止；当 K 个数中仅剩下一个数时，则直接移至下一运算层的同一位上作为“部份和”数；

第 4 步，在上述某位的相邻高位上，重复第 2 步及第 3 步的运算，直至 K 个运算数的每一位都已全部操作；当 K 个数的各位同时进行第 2 步及第 3 步运算时，则本步可跳越过去；

第 5 步，在下一个运算层中，将上述“按位和”数与进位行中的“进位数”进行前述第 2 步、第 3 步、第 4 步求和运算；

第 6 步，重复第 2 步至第 5 步的运算，直至不产生“混 Q 进位”为止，则最后一次“按位加”所得和数，即为所求混 Q 进制加法运算结果。

上述混 Q 进制数可以不另行编码；可以普通 8421 码等来编码；也可以全一码来编码。即，将各个混 Q 进制数的每一位数 S ，都以 $|S|$ 个 1 从最低位顺序至高位排列来对应，其余高位均为 0，总位数则为 $(Q-1)$ 位；同时，将混 Q 进制数中该位的数符，即表示该位为正为负，作为相应全一码中每一位上的数符。全一码编码混 Q 进制数时，二数加法仅为二数中 1 的不重复排列，称为“排 1”。

上述运算数可以是混 Q 进制数，或者普通对称 Q 进制数，或者混数数制数。

根据本发明的另一个方面，提供一种混 Q 进制、进位行处理器，包括：输入逻辑， K 重运算器，输出转换逻辑及控制器组成；其中， K 重运算器及控制器组成混 Q 运算控制逻辑；混 Q 进制数经全一码译码

移位寄存器输入逻辑至 K 重运算器； K 重运算器中，混 Q 进制数经 K 重运算获得混 Q 进制数的结果，经由译码器输出转换逻辑以混 Q 进制数或普通 Q 进制数、或普通十进制数通过输出逻辑输出，控制器协调控制整个运算控制的逻辑；其中，为 $2K$ 个寄存器中的每个寄存器的每一位分配一个符号位，该符号位为普通二态触发器；前面 K 个寄存器存放输入的 K 个混 Q 数，后面 K 个寄存器组成 K 个进位行；

在运算过程中，其中两个寄存器的某位，经累加器累加得到该位的和与高位进位，此进位送至任一进位寄存器的相邻高位；当下一个运算命令到达时，将进位行与原存数送入累加器相加；如此重复，最后再经累加器获得所求和数。当采用全一码编码时， K 重运算器中的累加器可以省略；

处理器中运算数为混 Q 进制数，或者普通对称 Q 进制数、或者混数数制数。

15 附图说明

图 1 是混 Q 进制计算机总逻辑框图。

图 2 是混 Q 进制、进位行处理器（运算控制）逻辑框图；

图 3 是 K 重运算器一位的逻辑框图；

图 4 是对冲逻辑（对冲器）的逻辑框图；

20 图 5 是划 Q 逻辑（划 Q 器）的逻辑框图；

具体实施方式

第一部分 混 Q 进制、进位行数字工程方法

1、《进位行方法》

25 1.1 进位与《进位行方法》

在电子计算机中，运算速度提高的关键之一，就在于“进位”。进位的获得，进位的存贮以及进位的参予运算都是至关重要的。“进位”就是争“速度”。在笔算中，还直接影响到“出错率”。

所谓《进位行方法》就是，在运算过程中，将产生的进位存放在参予运算与“按位和”数同等的位置上，然后与“按位和”一起进行运算。通常将同运算层中两数相加时，各位上的进位排列成一行，称为“进位行”。（运算层的概念，见下节）

举例如下，设两普通十进制数求和，算式以竖式求和。如式三：
为简化起见，这里将横竖式合写。个位运算 $(6+8)=14$ ，其进位

1

$$\begin{array}{r} 123456+345678=469134 \\ 345678 \\ \hline 468024 \dots\dots \oplus \text{ 行} \\ 111 \dots\dots \text{进位行} \\ \hline 469134 \end{array}$$

5

式三

写于下一行的高一位上。依此类推。式中二数相加时，各位上不计进位的求和，称为“按位加 \oplus ”。其和称为“按位和”。按位和的运算行，称为“ \oplus 行”。

10 各进位排成的行，称为“进位行”。由 \oplus 行与进位行组成“运算层”。

式中一些“+”号已省去。以后可以知道，在《混进方法 HJF》中，各个“运算层”只存在一种运算，这就是“+”。故可以不必在运算层中写出“+”号。

1.2 《进位行方法》分析

15

1.2.1 二数求和的分析

采用《进位行方法》的加法运算由上节可知：

- ① 两数相加时，每一位上只有二个数相加；
- ② 在进位行中直接标示进位，不存在任何困难；
- ③ 验算十分方便。

20

[引理一] 两数相加时，任意位上要么有进位记为 1，要么无进位

$$\begin{array}{r}
 5843029 + 4746979 = 10590008 \\
 \begin{array}{r}
 4746979 \\
 9589998 \\
 11\ 1111 \\
 \hline
 10590008
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \text{运算层}
 \end{array}$$

$$\begin{array}{r}
 5843029 \\
 + 4746979 \\
 \hline
 9589998 \\
 1\quad 1 \\
 \hline
 0589908 \\
 1\quad 1 \\
 \hline
 10589008 \\
 1 \\
 \hline
 10580008 \\
 1 \\
 \hline
 10590008
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{第一运算层} \\ \text{第二运算层} \\ \text{第三运算层} \\ \text{第四运算层} \end{array}$$

式五

式四

记为 0;

- [引理二] 两数相加时，任意位上的 \oplus 和可为 0~9 之一。但是，
5 当该位上有向高位进位时，该位上的 \oplus 和只能为 0~8 之一，而不能为 9。

由[引理一]和[引理二]可得：

[定理一] 两数相加时，当且仅当某位上没有向高位进位时，该位上的 \oplus 和才可能出现 9。

10 1.2.2 层次概念及运算层

设两数求和。算式为式四、式五

由式四可见，运算是分层次进行的，每一运算层，仅完成一项简单运算。这就是运算的“层次”概念，运算层将一个运算解剖成微运算、子运算。“层次”概念在数学中是基本概念。《进位行方法》正是建立在此概念基础上。以往的加法运算方法，本质上也隐含“层次”概念。因此，《进位行方法》中的“层次”从总体上看，并未增加运算的复杂性。反之，以往的方法由于隐含了“层次”，反而进一步增加了运算的复杂性。这一点，也进一步造成运算速度被明显降低。两者对比，就会一清二楚。

在《进位行方法》中，两数相加的各个运算层，可以合并为一个
20 运算层。如式五，请见进一步分析。

1.2.3 唯一的运算层

两数相加时，特别情况下会出现多次运算层。各层有如下关系成立。[引理三] 二数相加，当某位前一运算层上有进位时，其后各运

算层上均不可能出现进位。(由引理一、二得)

[引理四] 二数相加, 当某位后一运算层上有进位时, 其前各运算层上必无进位。(由引理一、二得)

231	786
786	666
+989	575
-----	321
866	699
111	+ 999
113	-----
2006	1616
	223
	12

	4046

式六 式七

[定理二] 二数相加时, 同一位各运算层上, 要么都无进位, 要么只能有一个进位。(由引理三、四得)

[推 论] 可以将全部各层进位行合并为一个进位行, 各运算层合并为一个运算层。

1.2.4 三数及三数以上求和分析

设三数求和, 算式为 $231+786+989=2006$ (见式六)

操作要点: ① “划Q”的运用;

所谓“划Q”, 即Q进位的两数在某位上相加时, 其按位加和为零, 但该位上产生进位(与两数符号一致)。进位放入进位行; 同时, 在某位上, 该两数均不再参加运算。在十进制时即为“划十”。

a、同一位上两数和为“十”时, 可在算式中将两数字以斜线划去, 然后在高位上补1。

b、同一位上几数和为20、30、40……等时, 可将几数字均划去, 然后在高位上补2、3、4……等。

又, 设六数求和。算式为 $786+666+575+321+699+999=2046$ (见式七)。

②多个数相加, 会出现二个及二个以上的运算层。为了减少运算层数, 同一位上的同一运算层空位中, 进位及⊕和数可以任意占位。

③尽量减少运算层。 a、较小的数, 直接合并算; b、尽量在“配对”中进位; c、尽量减少在第一运算层上相加数的个数, 尽量使第

二及二以上运算层不出现。

④同一位上，“相同数”、“连续数”等可直接获得“部分和”。

⑤设有 m 个数求和。（ m 为 ≥ 2 的自然数。）总运算层以 n 来表示。（ n 为非负整数）。则：

$$5 \quad \begin{cases} n_{\min}=0 \text{ (通常 } n=0, 1, 2, \text{ 而以 } n=1 \text{ 为最常见)} \\ n_{\max} = \begin{cases} m/2, & m \text{ 为偶数时} \\ m+1/2, & m \text{ 为奇数时} \end{cases} \end{cases}$$

式八

2、混数及混数数制

2.1《数制理论》

2.1.1 按同一种规则记录数，便于用来在一个数系统中进行运算的数的制度，称为“记数系统的制度”。简称为“数制”。一个数的质，首先就是由其所属的数制来决定的。恩格斯指出：“单个的数在记数法中已经得到了某种质，而且质是依照这种记数法来决定的。”

“一切数的定律都取决于所采用的记数法，而且被这个记数法所决定。”

15 《数制理论》就是研究数制的生成、分类、分析、比较、变换等以及数制在各邻近学科与实践应用的科学。它是数学的基础理论之一。数学科学，即“数”的科学。“数”的基本为“数制”。因此，“数制理论”是“数论”的基础，是“核心数学”的“核心”之一。

数制是数的属性。不存在没有所属数制的数，也不存在没有所属数的数制。[文中凡未标明数制的数，均指普通十进制数。下同。]

20 2.1.2 位值制数制

设，构造一个数系，其中的数由各不相同位置上的“数符”来表示。“数符”又称“数字”。数字通常从右向左水平排列，其相应的数值由低（小）到高（大）。每个数位上的数字给定一个单位值（又称“位值”）。由此来表示整个数系中每一个数的数制，称为“位值制数制”。

我们以下讨论的数制，都是“位值制数制”。简称为“数制”。所讨论的数均约定为整数。

2.1.3 数制的三大要素：数位 I ，数元集 Z_i 和权 L_i 。

a、数位 I ，表示数制中数的各位数字的位置。以 I （序数）从右

自左来表示。即, $i=1, 2, 3, \dots$ 表示该数的第 1, 2, 3, \dots 位。

b、数元集 Z_i , 表示第 i 位上的“数元”组成的集合。同一数制系统中, 各个数同一位上不同符号的全体, 组成一个该位上的数符集。该数符集中的元素, 称为“数的元素”。简称为“数元”。因此, 该数符集称为“数元集”。

数元集 Z_i 可以随着 i 的取值不同而不同, 也可以相同。当各位上的 Z_i 均为相同的 Z 时, 相应的数制称为“单一数制”; 当各位上的 Z_i 不全相同时, 相应的数制称为“混合数制”。单一数制为 Q 进制时, 称为“单一进制”; 混合数制为 Q 进制时, 称为“混合进制”。

数元集 Z_i 中的数元可为复数或其他多种多样符号。在《数制理论》中, 当定义空位表示 0, 具有隐含的“空位 0”, 即“空元”概念时, 则数元可为“空元”。以 a_j 来表示数元 (a_1, a_2, a_3, \dots) 以 ia_j 表示第 i 位上数元 a_j (j 为自然数), 数元集 Z_i 以集合 $\{a_1, \dots, a_j, \dots\}$ 来表示, 即 $Z_i = \{a_1, \dots, a_j, \dots\}$ 。

数元集 Z_i 的基数 P_i (P_i 为自然数) 表示了集的元素总数。它“不但决定它自己的质, 而且也决定其他一切数的质。” P_i 的取值不同, 标示了数元集 Z_i 的变化。各位上的 P_i 均相同, 则称为“单一基数”; 否则, 称为“混合基数”。

c、权 L_i , 表示第 i 位上的位值大小。特称此位值为“权 L_i ”。

L_i 为实数 (由于复数集非有序体, 故不采用)。不同的 L_i , 就决定了不同的位值。

在“编码理论”中, “编码”的主要特征就在于权 L_i 。

实际中常见的权 L_i 采用所谓“幂权”。即, 令 $L_i = Q_i^{(i-1)}$, Q_i 为实数。为便于计算起见, 取 Q_i 为自然数。常见各位 L_i 均为幂权, 而且成等比 Q 的数制。 Q 称为数制幂权的“底数”或数制的“底数”。底数 Q 的不同, 决定了不同的 L_i , 从而决定了不同的位值。通常, 称这种数制为“ Q 进制”。当 $Q=2, 3, 10$ 等时, 相应的数制就被称为“二进制”、“三进制”、“十进制”等。

另一种常用的权 L_i 采用“等权”, 即各位上的权相同。

在任一个 Q 进制数制中, 当 $P=Q$ 时, 自然数在该数制中可以连续唯一的形态表达, 称为“连续数制”, 又称“普通数制”;

当 $P > Q$ 时, 自然数在该数制中可以连续, 但有时以多种形态表

达，称为“重复数制”；

当 $P < Q$ 时，自然数在该数制中只能断续的形态表达，称为“断续数制”。

根据上述数制的三大要素，数制可以有无穷无尽的种类。

2.2 混数及混数数制

当数元集 Z_i 中，全部数元为连续整数时，该相应数制被称为“整数段数制”；

当数元集 Z_i 中，含数元 0 时，该相应数制被称为“含 0 数制”；

当数元集 Z_i 中，既有正数元，又有负数元时，相应数制被称为“混数数制”；混数数制中的数，称为“混数”。“混数”中既有正数元又有负数元的数，称“纯混数”。在 $\{Q\}$ 数中，既有正数元又有负数元的数，称为“纯 $\{Q\}$ 数”。（ $\{Q\}$ 定义见下一节。）

当数元集 Z_i 中，正负数元是相反数时，相应数制称为“对称数制”；显然，“对称数制”是“混数数制”的一种。

2.3 混 Q 进制 $\{Q\}$

在《数制理论》中，一个数制的名称采用“ $Z_i L_i$ ”。对 Q 进制，则为 $Z_i Q_i$ ；单一进制时，则为 ZQ 。例如 $\{0, 1, 2\}$ 三进制；或者 Z_i 以文字表明其特征。

对于含 0 的普通 Q 进制， $Z = \{0, 1, \dots, (Q-1)\}$ 。故 $ZQ = \{0, 1, \dots, (Q-1)\}Q$ ， Q 为 >1 的整数，称为“含 0 普通 Q 进制”。符号表示为 $\{\text{含 } 0, Q\}$ ；对于不含 0 的 $\{1, 2, \dots, Q\}Q$ ， Q 为自然数，称为“不含 0 普通 Q 进制”。符号表示为 $\{\text{不含 } 0, Q\}$ 。

含 0 和不含 0 的普通 Q 进制，合起来统称为“普通 Q 进制”， Q 为自然数。符号表示为 $\{Q\}$ 。当不致误解时，“含 0 普通 Q 进制”亦可称为“普通 Q 进制”，亦以符号 $\{Q\}$ 来表示。故可以符号 $\{=\}$ 及 $\{+\}$ 来表示普通二进制及普通十进制。

本文中的混数数制主要为以下几类。

对于含 0 的 $\{0, \pm 1, \dots, \pm (Q-1)\}Q$ 进制， Q 为 >1 的整数，称为“含 0 混 Q 进制”。符号表示为 $\{\text{含 } 0, Q^*\}$ ；对于不含 0 的 $\{\pm 1, \pm 2, \dots, \pm Q\}Q$ ， Q 为自然数，称为“不含 0 混 Q 进制”。符号表示为 $\{\text{不含 } 0, Q^*\}$ 。

含 0 和不含 0 的混 Q 进制，合起来统称为“混 Q 进制”， Q 为自

然数。符号表示为 $\{Q^*\}$ 。当不致误解时，“含 0 混 Q 进制”亦可称为“混 Q 进制”，亦以符号 $\{Q^*\}$ 来表示。故可以符号 $\{十^*\}$ 及 $\{二^*\}$ 来表示“混十进制”及“混二进制”。在《数制理论》中，它们的名称分别是：“单一基数 $P=19$ ，含 0，整数段，对称的十进制”。可写为 $\{十九，含 0，整数段，对称\}$ 十进制，或者写为 $\{0, \pm 1, \pm 2, \dots, \pm 9\}$ 十进制。一般情况下，进一步符号表示为 $\{+^*\}$ ，称为《混十进制》。此外，“单一基数 $P=3$ ，含 0，整数段，对称的二进制”。可写为 $\{三，含 0，整数段，对称\}$ 二进制，或者写为 $\{0, \pm 1\}$ 二进制。一般情况下，进一步符号表示为 $\{二^*\}$ ，称为《混二进制》。

在混数数制中，另一类为普通对称含 0 的 $\{0, \pm 1, \dots, \pm (Q-1)/2\}$ Q 进制， Q 为 > 1 的奇数，称为“含 0 普通对称 Q 进制”。符号表示为 $\{含 0, 称 Q\}$ ；对不含 0 的 $\{0, \pm 1, \dots, \pm Q/2\}$ Q 进制， Q 为正偶数，称为“不含 0 普通对称 Q 进制”。符号表示为 $\{不含 0, 称 Q\}$ 。

含 0 和不含 0 的普通对称 Q 进制，合起来统称为“普通对称 Q 进制”。 Q 为 > 1 的整数。符号表示为 $\{称 Q\}$ 。当不致误解时，“含 0 普通对称 Q 进制”，亦可称为“普通对称 Q 进制”，亦以符号 $\{称 Q\}$ 来表示。

3、《混进方法 HJF》及其混十进制 $\{十^*\}$ 四则运算。

采用混数和《进位行方法》来进行有理数运算的方法，称为《混数、进位行方法》，简称为《混进方法 HJF》。当用于算盘或笔算数字工程，采用的是 $\{+^*\}$ 混十进制等的《混进方法 HJF》。当用于电子计算机等之中时，采用的是 $\{二^*\}$ 混二进制及 $\{十^*\}$ 混十进制等的《混进方法 HJF》。

3. 1 $\{+^*\}$ 的加法

例 $1\bar{2}3 + 4\bar{5}\bar{6} = 427$ (见式九)

$$\begin{array}{r} 1\bar{2}3 \\ + 4\bar{5}\bar{6} \\ \hline 5\bar{7}\bar{3} \\ \hline 526 \\ \hline \bar{1}1 \\ \hline 427 \end{array}$$

式九

式中求得和为 $5\bar{7}\bar{3}$ 。当需要转化为普通十进制 $\{十\}$ 数时，和为

427. 一般来说, 所求和 $5\overline{73}$ 不必转化 (特别是作为计算过程中间结果时)。确需转化时, 方法见 4.1 转换法则。

3.2 { + * } 的减法

3.2.1 例 $1\overline{23}-4\overline{56}=1\overline{23}+4\overline{56}=\overline{339}$

5 例 $112+56-32-85+67-46=72$ (见式十)

$$\begin{array}{r} 2\overline{38} \\ \times 8\overline{9} \\ \hline 1\overline{872} \\ 1\overline{27} \\ 6\overline{38} \\ \hline 1\overline{26} \\ 1\overline{2102} \\ \hline \overline{6} \\ 1\overline{2502} \end{array}$$

$$\begin{array}{r} 1\overline{12} \\ 5\overline{6} \\ \overline{32} \\ \overline{85} \\ 6\overline{7} \\ + 4\overline{6} \\ \hline 1\overline{52} \\ \overline{72} \\ \hline 7\overline{2} \end{array}$$

式十一

式十

首先化为加法来运算, 这是由于混数的特性所决定。这一来, 实际计算中, 加减就合并为加法了。这就消除了通常连加减的困难。

10 3.2.2 约混。这是指二数求和时, 同一位上的相反数可以消去。也可称为“对消”或“对冲”。在算式中, 可以斜线划去。也就是说, 所谓“对冲”, 即两相反数, 其和为零。该某位上的两数不再参加以后的运算。在实际运算中, 采用先“对冲”后“划 Q”来获得混 Q 数的结果。

15 3.3 { + * } 的乘法

例 $2\overline{38} \times 8\overline{9} = 12\overline{502}$

(见式十一)

3.4 { + * } 的除法

$$\begin{array}{r} 2\overline{3} \overline{) 249} \\ \underline{46} \\ 112 \\ \underline{92} \\ 208 \\ \underline{207} \\ 1 \end{array}$$

式十二

$$\begin{array}{r} 5\overline{728} \\ \div 2\overline{3} \\ \hline 2 \overline{) 57} \\ \underline{46} \\ 112 \\ \underline{92} \\ 208 \\ \underline{207} \\ 1 \end{array}$$

式十三

$$\begin{array}{r} 3\overline{8} \overline{) 229} \overline{) 6} \\ \underline{31} \overline{) 270} \overline{) 6} \\ \underline{111} \quad 1 \\ 221716 \end{array}$$

式十四

例 $5728 \div 23 = 249 \dots 1$

要点：① 式十二采用原普通除法，现采用四则统一算式如式十三。

② 式十三中 $57 - 23 \times 2 = 57 + \overline{23} \times 2 = 57 + \overline{46}$ 也就是说，由于采用混数可使除法中的“减”过程变为“加”的过程。其余同此。

我们为了去掉“减”过程的思路，可以令被除数变号，然后，整个“减”过程完全变成“加”过程。这可使整个运算的复杂性进一步降低。

以后，我们的除法就以此来进行。但，应该注意，此时若出现余数则要将该余数变号后，才是最终运算结果的余数。

4、《混十进制》 $\{+^*\}$ 与《普通十进制》 $\{+\}$ 的关系。

4.1 $\{+^*\}$ 与 $\{+\}$ 数的转换法

这里指整数的情况，例如 $\{+^*\} 3\overline{8}2\overline{2}96 = \{+\} 221716$ (式十四)。

4.1.1 $\{+\}$ 数本身即为 $\{+^*\}$ 数的一种特况，故 $\{+\}$ 数不经转换即为 $\{+^*\}$ 数。

4.1.2 $\{+^*\}$ 数转换成 $\{+\}$ 。方法有几种：一种是将 $\{+^*\}$ 数变为一正一负的两个 $\{+\}$ 数求和。这有好多方式。其中，典型的是将该 $\{+^*\}$ 数中各正数字位及0位作为一正 $\{+\}$ 数，而将各负数字位作为一负 $\{+\}$ 数。例 $\{+^*\} 3\overline{8}2\overline{2}96 = \{+\} 302006 - 80290 = 221716$

再一种是在该数的各位上，使正数不变；负数的绝对值对10取“补”，同时在相邻的高位减1（即加 $\overline{1}$ ）。

$0 = \bar{0} = 00 = 000 = \dots = \dot{0} = 0_+$	$0 = \bar{0}\bar{0} = \bar{0}\bar{0}\bar{0} = \dots = \dot{0} = 0_-$
$1 = \bar{1} = 1\bar{9} = 1\bar{9}\bar{9} = \dots = 1\dot{9}$	$\bar{1} = \bar{1}\bar{9} = \bar{1}\bar{9}\bar{9} = \dots = \bar{1}\dot{9}$
$2 = \bar{2} = 1\bar{8} = 1\bar{9}\bar{8} = \dots = 1\dot{9}\bar{8}$	$\bar{2} = \bar{1}\bar{8} = \bar{1}\bar{9}\bar{8} = \dots = \bar{1}\dot{9}\bar{8}$
$3 = \bar{3} = 1\bar{7} = 1\bar{9}\bar{7} = \dots = 1\dot{9}\bar{7}$	$\bar{3} = \bar{1}\bar{7} = \bar{1}\bar{9}\bar{7} = \dots = \bar{1}\dot{9}\bar{7}$
$4 = \bar{4} = 1\bar{6} = 1\bar{9}\bar{6} = \dots = 1\dot{9}\bar{6}$	$\bar{4} = \bar{1}\bar{6} = \bar{1}\bar{9}\bar{6} = \dots = \bar{1}\dot{9}\bar{6}$
$5 = \bar{5} = 1\bar{5} = 1\bar{9}\bar{5} = \dots = 1\dot{9}\bar{5}$	$\bar{5} = \bar{1}\bar{5} = \bar{1}\bar{9}\bar{5} = \dots = \bar{1}\dot{9}\bar{5}$
$6 = \bar{6} = 1\bar{4} = 1\bar{9}\bar{4} = \dots = 1\dot{9}\bar{4}$	$\bar{6} = \bar{1}\bar{4} = \bar{1}\bar{9}\bar{4} = \dots = \bar{1}\dot{9}\bar{4}$
$7 = \bar{7} = 1\bar{3} = 1\bar{9}\bar{3} = \dots = 1\dot{9}\bar{3}$	$\bar{7} = \bar{1}\bar{3} = \bar{1}\bar{9}\bar{3} = \dots = \bar{1}\dot{9}\bar{3}$
$8 = \bar{8} = 1\bar{2} = 1\bar{9}\bar{2} = \dots = 1\dot{9}\bar{2}$	$\bar{8} = \bar{1}\bar{2} = \bar{1}\bar{9}\bar{2} = \dots = \bar{1}\dot{9}\bar{2}$
$9 = \bar{9} = 1\bar{1} = 1\bar{9}\bar{1} = \dots = 1\dot{9}\bar{1}$	$\bar{9} = \bar{1}\bar{1} = \bar{1}\bar{9}\bar{1} = \dots = \bar{1}\dot{9}\bar{1}$
$10 = \bar{1}\bar{0} = 1\bar{0} = 1\bar{9}\bar{0} = \dots = 1\dot{9}\bar{0}$	$\bar{1}\bar{0} = \begin{cases} \bar{1}\bar{0} = \bar{1}\bar{9}\bar{0} = \dots = \bar{1}\dot{9}\bar{0} \\ \bar{1}\bar{0} = \bar{1}\bar{9}\bar{0} = \dots = \bar{1}\dot{9}\bar{0} \end{cases}$
$11 = \bar{1}\bar{1} = 11 = 1\bar{9}\bar{1} = \dots = 1\dot{9}\bar{1}$	$\bar{1}\bar{1} = \bar{1}\bar{1} = 1\bar{9}\bar{1} = \dots = 1\dot{9}\bar{1}$
\vdots	\vdots

表一

另一种方法是：{+*} 数中，连续正数字（或 0）的数字段照写不变。如 $3 \times 2 \times \times 6$ 。但，当其不在 {+*} 数末尾（个位）时，则最低位加 $\bar{1}$ ；连续负数字的数字段，则使负数字的相反正数字与所求转换数字之和为 9，如 $\times 1 \times 70 \times$ 。然后，在其最低位加 1。

这样，求得结果为 221716，即为相应 {+} 数。

（注：式十四中连续负数字段右侧可划上分段线。当不致误解时，分段线可不划。）

4.2 {+*} 与 {+} 对照表及其说明（对照表见表一）

10 说明：表一中 $\bar{9}$ 表示为 9 的二次取负数（二次以上从略），余数同此。

① 表一中 0_+ 0_- 分别为从正负方向趋近于 0 所获得的 0；

② 表一中 $\dot{9}$ 表示任意非负整数位连续的 9，读作“延 9”。表中 $\dot{0}$ 表示任意非负整数位连续的 0，读作“延 0”。这种数，可以称为“无限延数”。

15 ③ 在 {+*} 数系统中，无限延数有且仅有 $(\dot{0}, \dot{0}, \dot{9}, \dot{9})$ 四种。由于 $\dot{0} = \dot{0}$ ，故无限延数有且仅有 $(\dot{0}, \dot{9}, \dot{9})$ 三种。亦可写为 $(\dot{0}, \pm \dot{9})$ 。

④ $\bar{0}=0$ ，由数 10 的两种表达形式可知。因此， $\bar{0}=0=\dot{0}=\ddot{0}$ 。

4.3 {+*} 与 {+} 关系分析

4.3.1 {+} 数是 {+*} 数的一部分，{+} 数集是 {+*} 数集的子集；

5 {+*} 数 \supset {+} 数，即 {+*} 数对 {+} 数有包含关系。

4.3.2 {+} 数与 {+*} 数的关系是“一多对应”关系，而不是“一一对应”关系。正由于此，{+*} 就获得了多样处理的灵活性。这是 {+*} 运算中多样性、快速性的原因。从这一点来说，{+*} 具有较强的功能。

10 它正是钱学森指出的数学第三层次“直接应用的工程技术”。这种“工程技术”与数字计算工程紧密结合的方法，称为“混 Q 进制、进位行数字工程方法”。

4.3.3 {+*} 数转换为 {+} 数，只能化为相应唯一的一个数。这是因为，{+*} 数可经 {+} 数加减直接获得，而 {+} 数加减运算后的结果是唯一的。反之，{+} 数也只能化为相应唯一的一组 {+*} 无限延数。所以，这种 {+} 数的“一”与 {+*} 无限延数的“一”组两者是“一一对应”关系。

由此，可建立一种 {+*} 数与 {+} 数的互为映射关系。

20 由于变换是集到自身上的对应，所以：{+} 与 {+*} 数是“一变换”。对于运算系统来说，{+} 与 {+*} 数系统是“自同构”。相应 {+} 数的各种运算性质，亦在 {+*} 数系统中成立。

4.3.4 {+*} 中 $P>Q$ ，因而在该数制中自然数有时会出现多种形态表达，这正是该数制灵活性所在，它使运算得以简便快捷。也可以说 {+*} 是以多样性来换取了灵活性。

25 {+} 中 $P=Q$ ，因而在该数制中，自然数是连续唯一形态表达。它没有这种多样性，也缺少了这种相应的灵活性。

可以这么说，本发明的关键正是在此。有了它，才有了《混进方法 HJF》，才有了“笔算工程”的新技术方案。有了它，也才有了处理器及其相应电子计算机新技术方案。

30 4.3.5 应当指出，显然，上述对 {+} 与 {+*} 的分析，完全相应于 {Q} 与 {Q*} 的分析，因为 {+} 与 {Q} 是同构的。由此可知，① {Q} 数与 {Q*} 数的关系是“一多对应”，而不是“一一对应”。②同时，{Q} 中的“一”

个数与相应的 $\{Q^*\}$ 中的“—”组无限延数，两者之间是“一一对应”关系。③ $\{Q\}$ 与 $\{Q^*\}$ 数系统是“自同构”。相应 $\{Q\}$ 数系统的各种运算性质，亦在 $\{Q^*\}$ 数系统中成立。

5、综合上述，可有如下简明结论：

- 5 混 Q 进制 $\{Q^*\}$ 及《混进方法 HJF》在数字工程中，可显著提高运算速度，而且大大降低笔算的出错率。它正是钱学森指出的数学第三层次“直接应用的工程技术”。这种“工程技术”与数字计算工程紧密结合的方法，称为“混 Q 进制、进位行数字工程方法”。

10 第二部分 混 Q 进制、进位行处理器

四则运算是一切运算的基础，显然也是电子计算机的基础。

图 1 为本发明处理器相应的混 Q 进制计算机总逻辑框图。由输入逻辑 101、CPU 中央处理器 102、外存 103、输出逻辑 104、控制台 105、输出转换逻辑 108 组成。中央处理器 102 由内存 106、混 Q 运算控制逻辑 107 组成。这些部件的连接关系是本领域公知的。其中，普通 Q 进制数通过输入逻辑 101 输入中央处理器 102，并通过混 Q 运算控制逻辑 107 进行混 Q 运算，运算结果连接输出转换逻辑 108，结果以混 Q 进制数、或普通 Q 进制数、或普通十进制数通过输出逻辑 104 输出。内存 106 及外存 103 与运算控制逻辑 107 交换数据，执行原有普通 Q 进制的程序。总操作由控制台 105 按既定程序控制，以时钟脉冲来实现。

图 2 为混 Q 进制、进位行处理器（运算控制）逻辑框图，由输入逻辑 101， K 重运算器 202，输出转换逻辑 108 及控制器 201 组成。其中，控制器 201 和 K 重运算器 202 组成混 Q 运算控制逻辑 107。

25 混 Q 进制数经全一码译码移位寄存器输入逻辑 101 至 K 重运算器 202； K 重运算器 202 中，混 Q 进制数经 K 重运算获得混 Q 进制数的结果，经由译码器输出转换逻辑 108 以混 Q 进制数、或普通 Q 进制数、或普通十进制数通过输出逻辑 104 输出。控制器 201 协调控制整个运算控制器的逻辑。

30 图 3 为 K 重运算器一位的逻辑框图，由第 I 位上的寄存器网 311、对冲网 312、划 Q 网 313 及累加器 304 组成。其中，累加器 304 为每一位带有一个正负符号位的普通累加器；寄存器网 311 由 A 寄存器 301、

B 寄存器 302、……、2K 寄存器 303 组成。对冲网 312 由一个对冲逻辑 305 巡检；或由 $K(2K-1)$ 个对冲逻辑 305、对冲逻辑 306、……、对冲逻辑 307 与寄存器网 311 中各个寄存器二二相连组成；或由分组、分级的对冲逻辑组成。划 Q 网 313 由一个划 Q 逻辑 308 巡检；或由 $K(2K-1)$ 个划 Q 逻辑 308、划 Q 逻辑 309、……、划 Q 逻辑 310 与寄存器网 311 中各个寄存器二二相连组成；或由分组、分级的划 Q 逻辑组成。

上述“K 重运算器”当 K 值较大时，可加以分级放大处理。

2K 个寄存器中，前面 K 个寄存器存放输入的 K 个混 Q 数。各个寄存器以及累加器的每位前附有符号位，该符号位为普通二态触发器。累加器仅为一个，它用来存放累加和数。符号位也可以放置在专用的符号位寄存器中，在运算时为存放混 Q 数的寄存器或累加器的每一位分配一个符号位。后面 K 个寄存器存放进位行数，组成 K 个进位行。

如果不采用对冲和划 Q，则在运算过程中，其中两个寄存器的某位，经累加器累加得到该位的和与高位进位，此进位送至任一进位行寄存器的相邻高位；当下一个运算指令到达时，将进位行与原存数送入累加器相加；如此重复，最后再经累加器获得所求和数。

为了加快运算速度，可采用对冲网和划 Q 网，由控制器或程序发出的指令，实施先“对冲”后“划 Q”运算，然后进行累加运算。当采用全一码编码时，K 重运算器 202 中的累加器 304 可以省略。

划 Q 产生的“进位”（与运算数同符号）送至 K 重运算器中任一进位行寄存器的相邻高位置“1”端。

图 4 为对冲逻辑（对冲器）的逻辑框图，由 A 寄存器的第 i 位 401，B 寄存器的第 i 位 402，同逻辑 403，异逻辑 404 及与门 405 组成。其中：A 寄存器的第 i 位 401，其前附有符号位，为普通二态触发器。Ai 的“1”端连接同逻辑 403 的输入、Ai 符的“1”端连接异逻辑 404 输入。

B 寄存器的第 i 位 402，其前附有符号位，为普通二态触发器。其 Bi 的“1”端连接同逻辑 403 的输入，Bi 符的“1”端连接异逻辑 404 的输入。同逻辑 403 的输出连接与门 405 输入；异逻辑 404 的输出连接与门 405 输入；与门 405 的输出连接 A 寄存器的第 i 位 401 的置“0”端及 B 寄存器第 i 位 402 的置“0”端。

图 5 为划 Q 逻辑（划 Q 器）的逻辑框图，由 A 寄存器的第 i 位 501，

B寄存器的第 i 位 502, Q 值判定逻辑 503, 同逻辑 504 及与门 505 组成, 其中: A寄存器的第 i 位 501, 其前附有符号位, 为普通二态触发器。A i 的 “1” 端连接 Q 值判定逻辑 503 的输入, A i 符的 “1” 端连接同逻辑 504 的输入。

- 5 B寄存器的第 i 位 502, 其前附有符号位, 为普通二态触发器。其 B i 的 “1” 端连接 Q 值判定逻辑 503 的输入; B i 符的 “1” 端连接同逻辑 504 的输入; Q 值判定逻辑 503 的输出连接与门 505 的输入; 同逻辑 504 的输出连接与门 505 输入; 与门 505 输出进位 (与运算数同符号) 送 K 重运算器中任一进位行寄存器的相邻高位置 “1” 端, 同时连接 A
- 10 寄存器的第 i 位 501 置 “0” 端及 B 寄存器第 i 位 502 的置 “0” 端。

当采用 {二} 运算时 (其他混数数制类似), 在运算及其控制中, 采用 $\{\bar{1}, 0, 1\}$ 三态进行。其中, $\bar{1}$ 、1 的正负号以一位 {二} 符号表示, 其权为 0。

- 当采用 {Q} 运算时, 运算器的输入不需要将 {Q} 数转换为 {Q*} 数,
- 15 因为 {Q} 数本来就是 {Q*} 数。即, $\{Q^*\} \text{数} = \{Q\} \text{数} + \text{纯 } \{Q^*\} \text{数}$ 。另一方面, 运算器的输出在一般中间过程也不必要将 {Q*} 数转换为 {Q} 数。只有在需要输出最终结果时, 才将 {Q*} 数转换为 {Q} 数 (实质是仅将纯 {Q*} 数转换为 {Q} 数)。这时, 本发明处理器相应的计算机在 “运算” 数字的输出界面上, 只需加上特别简单的 {Q*} 转换到 {Q} 译码器即可。这一点在技术上不存在任何困难。原则上, 本发明处理器相应的计算机其
- 20 外存及输入输出端与现有 {Q} 电子计算机完全一样 (包括程序在内)。这其中的原因就在于, 全部 {Q} 数本身均为 {Q*} 数所包含。在这种意义上, 现代 {Q} 数制电子计算机本来就是 {Q*} 电子计算机的特况。

- 本发明处理器相应的计算机系统中, 采用 “多重运算器”。例如,
- 25 采用 “八重运算器”。所谓 “八重运算器”, 即将 8 个数放入 8 个寄存器中, 一次性完成加减运算。设多重数为 K, 则 $K=2^n$ 较合适 (n 为自然数)。故 $K=2, 4, 8, \dots$ 。其中, 较重要的可能是 $K=8, 16, 256, 1024, 4096$ 等。同时, 乘法本质上原来就是连续加法, 除法本质上原来就是连续减法。因此, 在乘除中, 本发明处理器相应的计算机亦可
- 30 运用多重加减来处理。

本发明处理器相应的计算机除采用一般的累加器运算外, 为了加速运算可以采用 “对冲” 及 “划 Q” 逻辑。所谓 “对冲”, 即两相反

数相加，其和为零。所谓某位上的“划 Q”，即 Q 进制的两数相加时，其某位上按位加 \oplus 和为零，但产生进位（其符号与两数符号一致）。

“对冲”及“划 Q”逻辑线路在技术上是简单成熟的。见图 4 和图 5。

特别是，在 $\{Q^*\}$ 电子计算机中，仅仅只需先“对冲”后“划 Q”就能获得 $\{Q^*\}$ 数运算结果。当最终结果需要输出时，才将 $\{Q^*\}$ 数转换成 $\{十\}$ 数输出。当采用全一码编码时，K 重运算器中的累加器可以省略。

小结：

10 一、本发明处理器相应的计算机是混 Q 进制 $\{Q^*\}$ 的电子计算机，是《混进方法 HJF》电子计算机。

二、混 Q 进制 $\{Q^*\}$ 的电子计算机使现代以及未来基于其他原理上的各种电子计算机的运算速度大大提高。以八重运算器为例，粗略地估算将使运算速度提高五倍。也就是说，原 20 万次/s 的提高到 100 15 万次/s 左右；原 20 亿次/s 的提高到 100 亿次/s 左右。

第三部分 增 Q 进制 $\{Q^\Delta\}$ 及全一码

1. 增 Q 进制 $\{Q^\Delta\}$

1.1 定义及符号

20 在一个 Q 进制数制中，凡 $P=Q+1>Q$ 的进制，称为“增强 Q 进制”。简称为“增 Q 进制”，以符号 $\{Q^\Delta\}$ 来表示。Q 为自然数，显然， $\{0, 1, 2\}$ 二进制，即为“增二进制 $\{二^\Delta\}$ ”； $\{\bar{1}, 0, 1\}$ 二进制也就是混二进制 $\{二^*\}$ ，亦为“增二进制 $\{二^\Delta\}$ ”。此外，还有其他 $\{二^\Delta\}$ 。

25 1.2 增一进制 $\{一^\Delta\}$ 及其运算

增 Q 进制 $\{Q^\Delta\}$ 中，当 $Q = 1$ 时，即为增一进制 $\{一^\Delta\}$ 。增一进制 $\{一^\Delta\}$ 中，主要有二种。其一是 $\{0, 1\}$ 一进制，其元器件为二态器件。其二是 $\{\bar{1}, 1\}$ 一进制，其元器件亦为二态器件，它亦可表示全部整数。本文下面所称“增一进制 $\{一^\Delta\}$ ”均指 $\{0, 1\}$ 一进制。

30 增一进制 $\{一^\Delta\}$ 的运算。这里列出加法运算，例如 $\{+\} 4 + 3 + 2 = 9 = \{一^\Delta\} 110101 + 1011 + 101 = 11001100010101011$ 。

1.3 增一进制 $\{一^\Delta\}$ 与 $\{Q\}$ 的关系。

1.3.1 $\{-^{\Delta}\}$ 数与 $\{Q\}$ 数的转换法。

$\{-^{\Delta}\}$ 数转换成 $\{Q\}$ 数, 可以将 $\{-^{\Delta}\}$ 数中的各位数字 1, 以 $\{Q\}$ 计数即可。所得 $\{Q\}$ 计数和, 即为相应的 $\{Q\}$ 数。这就是说, $\{-^{\Delta}\}$ 数中有几个 1, 则相应的 $\{Q\}$ 数即为几。显然, 这是十分简单的法则。(见表二)

$\{Q\}$ 数转换成 $\{-^{\Delta}\}$ 数, 可将 $\{Q\}$ 数各位均乘以各位上的权, 然后将这些积以同样个数的 1, 分别在所要表达的 $\{-^{\Delta}\}$ 数位置上, 以不重复的方式列出即可。这就是说, $\{Q\}$ 数为几, 则 $\{-^{\Delta}\}$ 数中就有几个 1。显然, 这也是十分简单的法则。(见表三)

10 1.3.2 $\{-^{\Delta}\}$ 数与 $\{Q\}$ 数对照表及其说明

$\{+\}$	$\{二\}$	$\{-^{\Delta}\}$	$\{-^{\Delta}\}$	$\{二\}$	$\{+\}$
0	000	0...00000000 = 0 = 0	000	0	0
1	001	0...00000001 = 1 = 10	001	1	1
2	010	0...00000011 = 11 = 110 = 101 = 1010 = ...	010	1	1
3	011	0...00000111 = 111 = 1110 = 1101 = 11010 = ...	011	10	2
4	100	0...00001111 = 1111 = 11110 = 11101 = 111010 = ...	100	1	1
5	101	0...00011111 = 11111 = 111110 = 111101 = 1111010 = ...	101	10	2
6	110	0...00111111 = 111111 = 1111110 = 1111101 = 11111010 = ...	110	10	2
7	111	0...01111111 = 1111111 = 11111110 = 11111101 = 111111010 = ...	111	11	3
	=	=	=	=	=

表 三

表 二

1				
1	1			
1	2	1		扬
1	3	3	1	辉
1	4	6	4	1 三
.	.			角
.	.			形

表四

见表二、三 (令 $Q=2, 10$)

说明: ① $\{-^{\Delta}\}$ 数可表示全部 $\{Q\}$ 数

②有较多的重复数, 以 4 位 $\{-^{\Delta}\}$ 数为例, 除 0 及 4 唯一外,

其余均有重复数。其中，1有4个；2有6个；3有4个。于是，从0~4的重复数分别为1, 4, 6, 4, 1个。这与二项式展开系数 C_n^k 是一致的。(位数 n 为自然数， k 为0~ n 。)(见表四 扬辉三角形。)

③表中0表示为任意非负整数位连续的0。这与混 Q 进制中是一样的。称为“无限延数”。 $\{-^{\Delta}\}$ 数中，无限延数有且仅有一个，即为“0”。

1.3.3 $\{-^{\Delta}\}$ 与 $\{Q\}$ 关系分析。

(1) $Q \supset 1$, Q 为自然数；1为最小的自然数，也是最基本的自然数单元。 Q 包含1，这使得相应的 $\{Q\}$ 及 $\{-^{\Delta}\}$ 之间存在自然的联系。

(2) $\{Q\}$ 数与 $\{-^{\Delta}\}$ 数的关系是“一多对应”关系，而不是“一一对应”关系。正由于此， $\{-^{\Delta}\}$ 就获得了多样处理的灵活性。这是 $\{-^{\Delta}\}$ 运算中快速性的原因之一。从这一点来说， $\{-^{\Delta}\}$ 具有较强的功能。

(3) $\{-^{\Delta}\}$ 数转换为 $\{Q\}$ 数，只能化为相应唯一的一个数。这是因为， $\{-^{\Delta}\}$ 数可经 $\{Q\}$ 加减直接获得，而 $\{Q\}$ 数加减运算后的结果是唯一的。反之， $\{Q\}$ 也只能化为相应唯一的一组 $\{-^{\Delta}\}$ 无限延数。所以，这种 $\{Q\}$ 数的“一”与 $\{-^{\Delta}\}$ 无限延数的“一”组两者是“一一对应”关系。由此，可建立一种 $\{-^{\Delta}\}$ 数与 $\{Q\}$ 数的互为映射关系。对于运算系统来说， $\{Q\}$ 与 $\{-^{\Delta}\}$ 数系统是“同构”。相应 $\{Q\}$ 数的各种运算性质，亦在 $\{-^{\Delta}\}$ 数系统中成立。

(4) $\{-^{\Delta}\}$ 中 $P=Q+1$ ， Q ，因而在该数制中，自然数有时会出现多种形态表达，这正是该数制灵活性所在，它使得运算得以简便快捷。也可以说， $\{-^{\Delta}\}$ 是以多样性来换取了灵活性。

$\{Q\}$ 中 $P=Q$ ，因而在该类数中，自然数是连续唯一形态表达。它没有这种多样性，也缺少了这种相应的灵活性。

(5) 上述 $\{-^{\Delta}\}$ 与 $\{Q^*\}$ 相结合，使得功能更加增强。考虑到 $\{-^{\Delta}\} \rightarrow \{Q\} \rightarrow \{Q^*\}$ ，这其中有着内在的联系。显然，这一切均在预料之中。

1.4 增一进制 $\{-^{\Delta}\}$ 的应用

1.4.1 增一进制 $\{-^{\Delta}\}$ 的运算是一种优异的运算。由于它以么元1配以0构造数，而且权为1，故其运算常以“传送”来实现。这

是 $\{-^{\Delta}\}$ 数运算中快速性原因之一。 $\{-^{\Delta}\}$ 数运算中的“进位”，也可以当前位的二数按位加和为 0，而进位为 Q 的“划 Q”逻辑实现。这种“传送”及“划 Q”的逻辑实现，结构特别简单，速度却特别的快。这是 $\{-^{\Delta}\}$ 数运算中快速性原因之二。

5 当 $\{-^{\Delta}\}$ 数与 $\{Q^{\ast}\}$ 数结合运算时，又补充了“对冲”这一结构更为简单、速度更为快速的逻辑。这是 $\{-^{\Delta}\}$ 数运算快速性原因之三。

1.4.2 $\{-^{\Delta}\}$ 与 $\{Q^{\ast}\}$ 结合可作为多种新一代超高速电子计算机的技术方案。[详见第四部分。]

10 2. 全一进制、全一数及全一码

2.1 全一进制和全一数

增一进制 $\{-^{\Delta}\}$ 数的多样性是 $\{-^{\Delta}\}$ 数运算快速的原因之一。但是，由于 $\{-^{\Delta}\}$ 数具有极端的多样，常造成数的表达形式难以把握。由此造成运算数过于分散，不便于控制，势必影响运算速度。

15 因此，在一般情况下，有必要对 $\{-^{\Delta}\}$ 数加以某种约束条件。这就产生了“全一进制”。

在增一进制 $\{-^{\Delta}\}$ 的正整数中，限定每一组无限延数，只选取从个位开始，从右向左连续排列么元 1 的唯一的一种形态表达，高位上均为 0，可以空位表示。例如： $\{+\}$ 数 $3 = \{-^{\Delta}\}$ 数 $111/1110/110$
20 $1/\dots$ (“/”表“或者”)，限定为 $\{+\}$ $3 = \{-^{\Delta}\}$ $0111 = \{-^{\Delta}\}$ 111 。这样，每一组无限延数中的重复数均被删除，只剩下一个全是 1 的唯一形态。我们称为“全一数”。表达“全一数”的进制称之为“全一进制”。表二中， $\{-^{\Delta}\}$ 数最左边的形态，即为“全一进制”数。当考虑到正负整数时，可以将该全一进制数的正负符号，分配到该数的各位上去。从而构造带符号的全一进制。下述“全一进制”均为此种带符号的全一进制。

因此，“全一进制”可以是加特定约束条件的增一进制 $\{-^{\Delta}\}$ 。

在《数制理论》中，当定义空位表示 0，具有隐含的“空位 0”，即“空元”概念时，则全一进制可以从“加符号位的 $\{1\}$ 一进制”获得；全一进制也可以从不含 0 的混 Q 进制 $\{\text{不含 } 0, Q^{\ast}\}$ 中的“ $\{\bar{1}, 1\}$ 一进制”加约束条件获得，约束条件为该进制数必须各位上符号均相同；全一进制还可以从不含 0 增一进制 $\{\text{不含 } 0, -^{\Delta}\}$ 中的“ $\{\bar{1}, 1\}$

一进制”加上上述同样约束条件获得。

2.2 全一码

全一进制显然具有如下优缺点。优点：①运算速度快。“传送”代替了“翻转”。②多重运算时，不需要二、二求和，只需要先“对冲”及后“划Q”即可得结果。这就大大加快了总体运算速度。③与{Q}转换方便。缺点：①“字长”太长，位数多。当取可变字长时，其平均字长仅为一半。②荷载信息量较小。因此，根据全一进制的优缺点，扬长避短，以全一进制来编码{Q*}是合适的。以“全一进制”来编码，称为“全一编码”。“全一编码”中采用的“全一数”，称为“全一码”。由上述全一进制是带符号的可知，全一码也是带符号的。表五，显示出全一码一位，编码{二}数元的情况。由表五可见，全一码一位编码的{二}数，即为{二}数本身。表六，显示出以全一码九位，编码{十}数元的情况。由表六可见，全一码九位编码的{十}，字长增加至9倍。当取可变字长时，其平均字长仅为5倍。例如：{十}

23 = 全一码 = 。

一位全一码{二}

0	0
1	1

表五

九位全一码{十}

00 ... 0	0
00 ... 1	
00 ... 11	
...	
111111111	9

表六

对于混Q进制{Q*}，则可以全一码来编码。需要指出的是，这里全一码一位编码的{二*}数，即为{二*}数本身；这里{十*}数，则以九位全一码来编码。

2.3 全一码的计算。

全一码的计算非常简单。以二数加法为例，仅为二数中1的不重复排列，称为“排1”。如11+111=11111。特别是，在{Q*}数字工程中，仅仅只需先“对冲”后“划Q”就能获得{Q*}数运算结果。当最终结果需要输出时，才将以全一码编码的{Q*}数转换成{十}数输出。

2.4 全一码的应用。

全一码主要应用于对 $\{Q\}$ 及 $\{Q^*\}$ 数进行编码。特别是,

①采用全一码九位编码 $\{十\}$ 数, 可以实现普通十进制 $\{十\}$ 、全一码处理器。

②采用全一码九位编码 $\{十^*\}$ 数, 可以实现混十进制 $\{十^*\}$ 、全一码处理器。

③采用全一编码 $\{Q^*\}$ 数, 可以实现混 Q 进制 $\{Q^*\}$ 、进位行、全一码处理器。

④采用全一码九位编码 $\{+\}$ 或 $\{+^*\}$ 数, 再以正负码来二次编码, 可以实现又一种新型算盘。

⑤采用全一码九位编码 $\{+\}$ 或 $\{+^*\}$ 数, 再以正负码来二次编码, 可以实现又一种新型笔算工程。

第四部分 混十进制 $\{十^*\}$ 、全一码、进位行电子计算机

(一) 人类历史中, $\{十\}$ 计算应用的广度和深度, 均是其他进制所不能比拟的。人类长期历史文化的底蕴与沉积, 使得 $\{十\}$ 具有牢不可破的至尊地位。因此, 混十进制 $\{十^*\}$ 电子计算机具有特别重要的意义。为此, 本部分将混十进制 $\{十^*\}$ 、全一码、进位行电子计算机特别列出。

(二) 混十进制 $\{十^*\}$ 、全一码、进位行电子计算机, 是全一码九位来编码的 $\{十^*\}$ 电子计算机。

(三) 该电子计算机的运算采用《混进方法 HJF》。即, 混十进制 $\{十^*\}$ 的《混进方法 HJF》。

(四) 混十进制 $\{十^*\}$ 、全一码、进位行电子计算机总逻辑框图, 如图 1 所示。

混十进制 $\{十^*\}$ 、全一码、进位行电子计算机中, $\{十^*\}$ 数以全一码九位来编码输入及最终输出转换为普通十进制数。

当采用 $\{十^*\}$ 运算时, 运算器的输入不需要将 $\{十\}$ 数转换为 $\{十^*\}$ 数, 因为 $\{十\}$ 数本来就是 $\{十^*\}$ 数。即, $\{十^*\}$ 数 = $\{十\}$ 数 + 纯 $\{十^*\}$ 数。另一方面, 在一般中间过程, 运算器的输出, 也不必要将 $\{十^*\}$ 数转换为 $\{十\}$ 数。只有在需要输出最终结果时, 才将 $\{十^*\}$ 数转换为 $\{十\}$ 数 (实质是仅将纯 $\{十^*\}$ 数转换为 $\{十\}$ 数)。这时, 本发明的计算机在“运算”的输出界面上, 只需加上特别简单的全一码编码

的{十*}转换到{十}的译码器即可。这一点在技术上不存在任何困难。原则上，本发明的计算机其内外存及输入输出端与现有{十}电子计算机完全一样（包括程序在内）。这其中的原因就在于，全部{十}数本身均为{十*}数所包含。在这种意义上，现代{十}数制电子计算机本来就是{十*}电子计算机的特况。

（五）混十进制{十*}、全一码、进位行电子计算机系统中，采用“多重运算器”。例如，采用“八重运算器”。所谓“八重运算器”，即将8个数放入8个寄存器中，一次性完成加减运算。设多重数为K，则 $K=2^n$ 较合适（n为正整数）。故K=2、4、8、...。其中，较重要的是K=8、16、256、1024、4096等。同时，乘法本质上原来就是连续加法，除法本质上原来就是连续减法。因此，在乘除中，本发明的计算机亦可运用多重加减来处理。

（六）混十进制{十*}、全一码、进位行电子计算机采用“对冲”及“划十”逻辑。所谓“对冲”，即两相反数相加，其和为零。所谓某位上的“划十”，即十进制的两数相加时，其某位上按位加 \oplus 和为零，但产生进位（其符号与两数符号一致）。“对冲”及“划十”逻辑线路在技术上是简单成熟的。见图4和图5。

特别是，在{十*}全一码、进位行电子计算机中，仅仅只需先“对冲”、后“划十”，就能获得运算结果。当最终结果需要输出时，才将全一码编码的{十*}数转换成{十}数输出。

（七）小结：

一、{十*}全一码、进位行电子计算机是混十进制{十*}的电子计算机，是《混进方法HJF》电子计算机。

二、本发明的{十*}全一码、进位行电子计算机使现代以及未来基于其他原理上的各种电子计算机运算速度大大提高。以八重运算器为例，粗略地估算将使运算速度提高五倍以上。也就是说，原20万次/s的提高到100万次/s左右；原20亿次/s的提高到100亿次/s左右。

权 利 要 求

1. 一种混 Q 进制、进位行数字工程方法，包括以下步骤：

第 1 步，设，参予运算的数为 K 个普通 Q 进制数，为 $K \geq 2$ 的正整数， Q 为自然数；将这些普通 Q 进制数的正负符号，分配到相应这些数的每一位上去。

第 2 步，对 K 个数同时进行混 Q 进制的求和运算，从最低位开始或各位同时按位相加，即在某一位上，取前述 K 个数中的二个数按位相加，得到“按位和”为该位这二个数相加的和数，将此和数记入下一运算层，作为“部份和”数；同时所得“混 Q 进位”，则存放到下一运算层的任一进位行中与该位相邻的高位处；

第 3 步，在该位上取 K 个数中的另二个数，进行第 2 步的运算，如此反复，直至 K 个数均取完为止；当 K 个数中仅剩下一个数时，则直接移至下一运算层的同一位上作为“部份和”数；

第 4 步，在上述某位的相邻高位上，重复第 2 步及第 3 步的运算，直至 K 个运算数的每一位都已全部操作；当 K 个数的各位同时进行第 2 步及第 3 步运算时，则本步可跳越过去；

第 5 步，在下一个运算层中，将上述“按位和”数与进位行中的“进位数”进行前述第 2 步、第 3 步、第 4 步求和运算；

第 6 步，重复第 2 步至第 5 步的运算，直至不产生“混 Q 进位”为止，则最后一次“按位加”所得和数，即为所求混 Q 进制加法运算结果。

2. 如权利要求 1 的混 Q 进制、进位行数字工程方法，其特征在于运算为“混 Q 进制”运算；其中， $\{0, \pm 1, \dots, \pm (Q-1)\}Q$ 进制， $P > Q$ ， Q 为 >1 的整数，称为“含 0 混 Q 进制”； $\{\pm 1, \dots, \pm (Q-1)\}Q$ 进制， $P > Q$ ， Q 为自然数，称为“不含 0 混 Q 进制”；当不致误解时，“混 Q 进制”即指“含 0 混 Q 进制”。

3. 如权利要求 1 的混 Q 进制、进位行数字工程方法，其特征在于运算采用“进位行方法”；即在运算过程中，将产生的进位存放在相邻高位“进位行”中，然后与“按位和”一起进行运算。

4. 如权利要求 1-3 任一个的混 Q 进制、进位行数字工程方法，其特征在于在某一位上，对 K 个数中的二个数进行求和运算时，如果其

中两个运算数的该位为相反数，则该位和为零，然后将该两个运算数的某位均以逻辑方式置“0”，不再参加以后的运算，这称为“对冲”；如果在某一位上，对 K 个数中的二个数进行求和运算时，其中两个运算数的按位加和为零，但产生进位，则将其进位放入任一进位行中的相邻高位，然后将该两个运算数的某位均以逻辑方式置“0”，不再参加以后的运算，这称为“划Q”；或者，不采用“对冲”及“划Q”。

5 5. 如权利要求 1-4 任一个的混 Q 进制、进位行数字工程方法，其特征在于可以不编码混 Q 进制；可以普通 8421 码等来编码混 Q 进制数；也可以全一码来编码混 Q 进制数，即将各个混 Q 进制数的每一位数 S ，都以 $|S|$ 个 1 从最低位顺序至高位排列来对应，其余高位均为 0，总位数则为 $(Q-1)$ 位；同时，将混 Q 进制数中该位的数符，即表示该位的数为正或负，作为相应全一码中每一位上的数符。

6. 如权利要求 1-5 任一个的混 Q 进制、进位行数字工程方法，其特征在于当采用全一码来编码混 Q 进制数时，二数加法仅为二数中 1 的不重复排列，称为“排 1”。

7. 权利要求 1-4 任一个的混 Q 进制、进位行数字工程方法，其中所述运算数是混 Q 进制数， Q 为自然数；或者是普通对称 Q 进制数， Q 为 >1 的整数；或者是混数数制数。

8. 一种混 Q 进制、进位行处理器，包括：输入逻辑 (101)， K 重运算器 (202)，输出转换逻辑 (108) 及控制器 (201) 组成；其中， K 重运算器 (202) 及控制器 (201) 组成混 Q 运算控制逻辑 (107)；混 Q 进制数输入经全一码译码移位寄存器输入逻辑 (101) 至 K 重运算器 (202)； K 重运算器 (202) 中，混 Q 进制数经 K 重运算获得混 Q 进制数的结果，经由译码器输出转换逻辑 (108) 以混 Q 进制数、或普通 Q 进制数、或混数数制数、或普通十进制数通过输出逻辑 (104) 输出，控制器 (201) 协调控制整个运算控制器的逻辑；其中，为 $2K$ 个寄存器中的每个寄存器以及累加器的每一位分配一个符号位，该符号位为普通二态触发器；前面 K 个寄存器存放输入的 K 个混 Q 数，后面 K 个寄存器组成 K 个进位行；

30 在运算过程中，其中两个输入寄存器的某位，经累加器累加得到该位的和与高位进位，此“部份和”数送至输入寄存器中，替换原存数；此进位送至任一进位行寄存器的相邻高位；当下一个运算指令到

达时, 将进位数与“按位和”数送入累加器相加;

如此重复, 最后再经累加器获得所求和数。

9. 如权利要求 8 的混 Q 进制、进位行处理器, 进一步包含:

5 对冲网 (312) 和划 Q 网 (313) 与寄存器网 (311) 中各个寄存器
二二相连组成; 或者, 不采用对冲网 (312) 和划 Q 网 (313);

当采用“对冲”及“划 Q”时, 由控制器或程序发出的指令, 先对
某位上运算数的各个数实施“对冲”及“划 Q”运算, 然后进行累加运
算, 其中累加器 (304) 为每一位带有一个符号位的普通累加器; 当采
用全一码编码时, K 重运算器 (202) 中的累加器 (304) 可以省略;

10 某位上划 Q 产生的“进位”, 送至 K 重运算器 (202) 中任一进位行
寄存器的相邻高位置“1”端。

10. 如权利要求 9 的混 Q 进制、进位行处理器, 其中的对冲网 (312)
由一个对冲逻辑 (305) 巡检; 或由 K (2K-1) 个对冲逻辑 (305、
306、...、307) 与寄存器网 (311) 中各个寄存器二二相连组成; 或
15 由分组、分级的对冲逻辑组成;

其中的划 Q 网 (313) 由一个划 Q 逻辑 (308) 巡检; 或由 K (2K-1)
个划 Q 逻辑 (308、309、...、310) 与寄存器网 (311) 中各个寄存器
二二相连组成; 或由分组、分级的划 Q 逻辑组成;

上述“K 重运算器”当 K 值较大时, 可以进行分级放大。

20 11. 如权利要求 10 的混 Q 进制、进位行处理器, 其中的对冲逻辑
由 A 寄存器的第 i 位 (401), B 寄存器的第 i 位 (402), 同逻辑 (403),
异逻辑 (404) 及与门 (405) 组成; 其中: A 寄存器的第 i 位 (401),
其前附有符号位, 为普通二态触发器; 其 A_i 的“1”端连接同逻辑 (403)
的输入、 A_i 符的“1”端连接异逻辑 (404) 输入; B 寄存器的第 i 位
25 (402), 其前附有符号位, 为普通二态触发器; 其 B_i 的“1”端连接
同逻辑 (403) 的输入, B_i 符的“1”端连接异逻辑 (404) 的输入;
同逻辑 (403) 的输出连接与门 (405) 输入; 异逻辑 (404) 的输出连
接与门 (405) 输入; 与门 (405) 的输出连接 A 寄存器的第 i 位 (401)
的置“0”端及 B 寄存器第 i 位 (402) 的置“0”端;

30 其中的划 Q 逻辑由 A 寄存器的第 i 位 (501), B 寄存器的第 i 位
(502), Q 值判定逻辑 (503), 同逻辑 (504) 及与门 (505) 组成;
A 寄存器的第 i 位 (501), 其前附有符号位, 为普通二态触发器; 其

5 A_i 的“1”端连接 Q 值判定逻辑 (503) 的输入, A_i 符的“1”端连接同逻辑 (504) 的输入; B 寄存器的第 i 位 (502), 其前附有符号位, 为普通二态触发器; 其 B_i 的“1”端连接 Q 值判定逻辑 (503) 的输入; B_i 符的“1”端连接同逻辑 (504) 的输入; Q 值判定逻辑 (503) 的输出连接与门 (505) 的输入; 同逻辑 504 的输出连接与门 (505) 输入; 与门 (505) 输出进位 (同符号) 送至 K 重运算器中任一进位行寄存器的相邻高位置“1”端, 同时连接 A 寄存器的第 i 位 (501) 置“0”端及 B 寄存器第 i 位 (502) 的置“0”端。

10 12. 如权利要求 8 的混 Q 进制、进位行处理器, 其中所述运算数用全一码编码来表示; 或者, 以普通 8421 码来编码; 或者, 不编码。

13. 如权利要求 8 的混 Q 进制、进位行处理器, 其中所述运算数是混 Q 进制数, Q 为自然数; 或者是普通对称 Q 进制数, Q 为 >1 的整数; 或者是混数数制数。

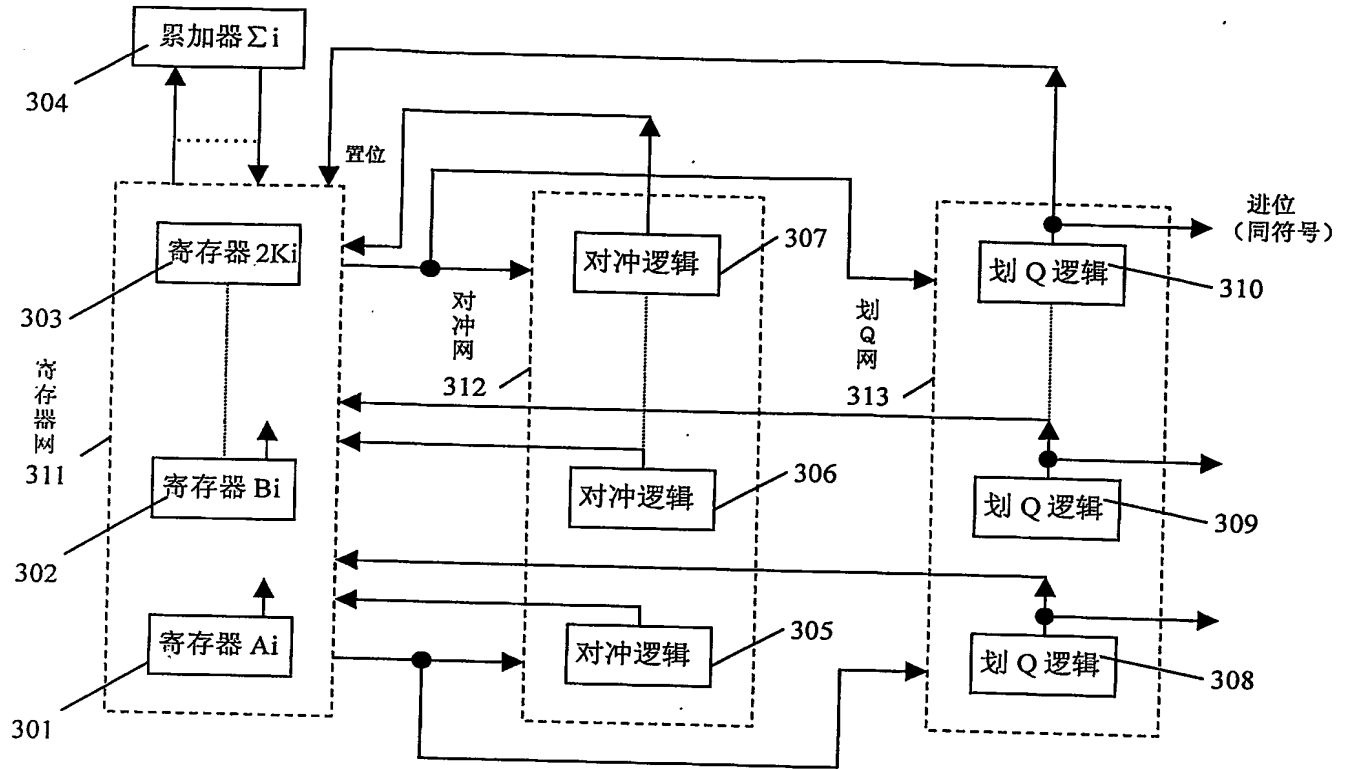


图 3

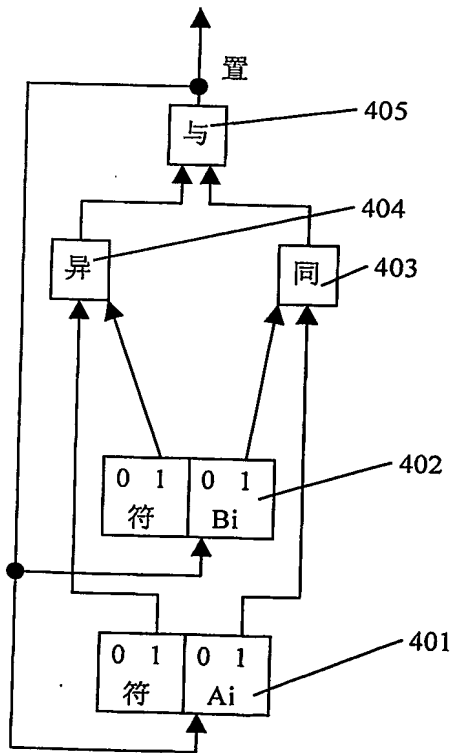


图 4

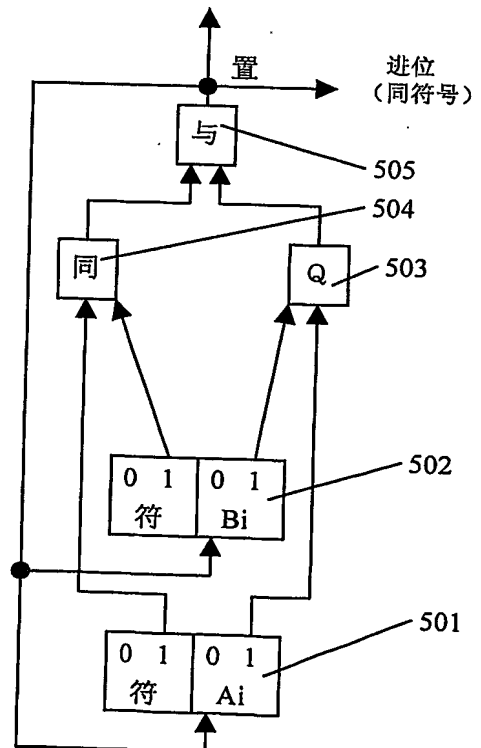


图 5